Руководство пользователя KAlgebra

Aleix Pol
Перевод на русский язык: Николай Шафоростов
Дополнение и исправление перевода на русский язык:
Максим Воробьёв
Дополнение и исправление перевода на русский язык:
Олеся Герасименко
Дополнение и исправление перевода на русский язык:
Мария Шикунова

Оглавление

1 Введение 6

2 Синтаксис 8

3 Использование калькулятора 9

4 Двумерные графики 11
- 4.1 Синтаксис . . . 11
- 4.2 Возможности . . . 11

5 Трёхмерные графики 13

6 Словарь 15

7 Команды, поддерживаемые KAlgebra 16
- 7.1 plus . . . 16
- 7.2 times . . . 16
- 7.3 minus . . . 16
- 7.4 divide . . . 16
- 7.5 quotient . . . 17
- 7.6 power . . . 17
- 7.7 root . . . 17
- 7.8 factorial . . . 17
- 7.9 and . . . 17
- 7.10 or . . . 17
- 7.11 xor . . . 18
- 7.12 not . . . 18
- 7.13 gcd . . . 18
- 7.14 lcm . . . 18
- 7.15 rem . . . 18
- 7.16 factorof . . . 18
- 7.17 max . . . 19
- 7.18 min . . . 19
- 7.19 lt . . . 19
- 7.20 gt . . . 19
- 7.21 eq . . . 19
- 7.22 neq . . . 19

7.23 leq . . . 20
7.24 geq . . . 20
7.25 implies . . . 20
7.26 approx . . . 20
7.27 abs . . . 20
7.28 floor . . . 21
7.29 ceiling . . . 21
7.30 sin . . . 21
7.31 cos . . . 21
7.32 tan . . . 21
7.33 sec . . . 21
7.34 csc . . . 22
7.35 cot . . . 22
7.36 sinh . . . 22
7.37 cosh . . . 22
7.38 tanh . . . 22
7.39 sech . . . 22
7.40 csch . . . 23
7.41 coth . . . 23
7.42 arcsin . . . 23
7.43 arccos . . . 23
7.44 arctan . . . 23
7.45 arccot . . . 23
7.46 arccosh . . . 24
7.47 arccsc . . . 24
7.48 arccsch . . . 24
7.49 arcsec . . . 24
7.50 arcsech . . . 24
7.51 arcsinh . . . 24
7.52 arctanh . . . 25
7.53 exp . . . 25
7.54 ln . . . 25
7.55 log . . . 25
7.56 conjugate . . . 25
7.57 arg . . . 25
7.58 real . . . 26
7.59 imaginary . . . 26
7.60 sum . . . 26
7.61 product . . . 26
7.62 diff . . . 26
7.63 card . . . 26
7.64 scalarproduct . . . 27
7.65 selector . . . 27
7.66 union . . . 27
7.67 forall . . . 27
7.68 exists . . . 27
7.69 map . . . 27
7.70 filter . . . 28
7.71 transpose . . . 28

8 Авторские права и лицензия 29

Аннотация

KAlgebra — это программа-калькулятор с функцией построения графиков. Позволяет вычислять значения выражений и строить двумерные и трёхмерные графики функций. В основе KAlgebra лежит язык математической разметки MathML, однако его знания не требуется для использования программы KAlgebra.

Глава 1

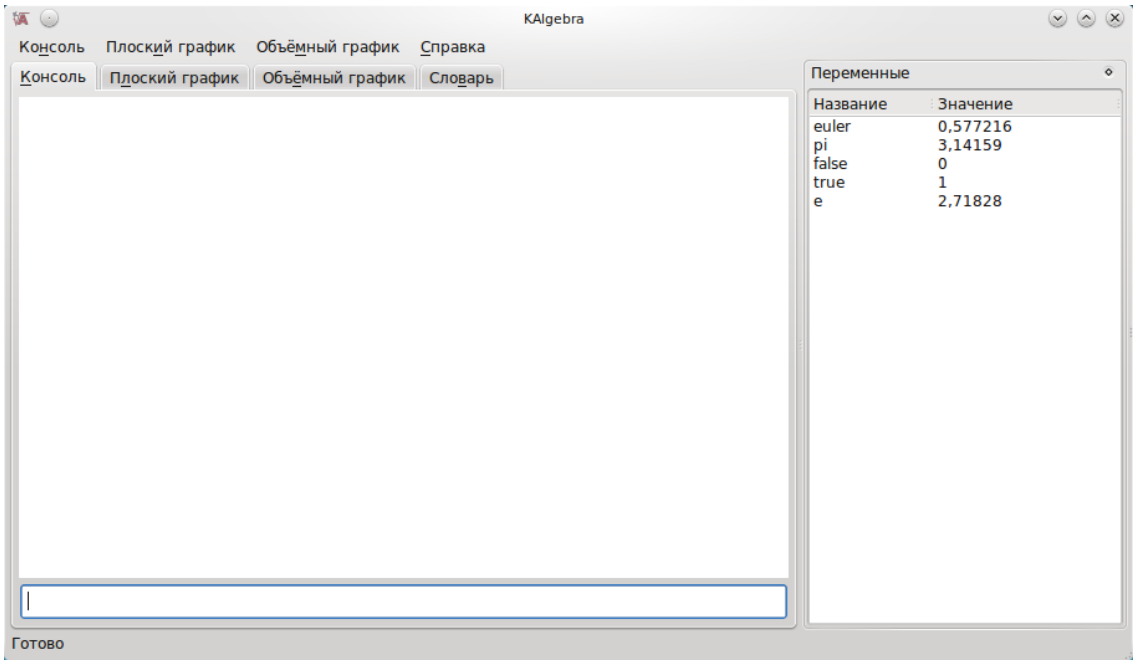
Введение

С помощью KAlgebra возможно выполнять различные математические операции и строить графики. Программа позволяет решать как простые, так и сложные задачи, при этом пользователю не требуется знать язык MathML или обладать глубокими познаниями в математике.

Среди возможностей программы:

- Калькулятор для быстрого и простого упрощения математических функций.
- Возможность создания сценариев для сложных последовательностей вычислений.
- Языковые возможности, в том числе определение функций и автодополнение синтаксиса.
- Возможности исчисления, в том числе символическое дифференцирование, векторное исчисление и управление списками.
- Построение графиков функций с показом координат при наведении курсора для поиска корней графическим способом и другие типы анализа.
- Построение объёмных графиков для визуализации трёхмерных функций.
- Встроенный словарь операций для быстрого поиска справочной информации о доступных функциях.

Ниже представлено окно программы KAlgebra:



Главное окно KAlgebra состоит из четырёх вкладок: Калькулятор, Плоский график, Объёмный график и Словарь. Под каждой вкладкой находится поле для ввода функций и выполнения вычислений, а также поле, в котором отображаются результаты.

Для управления сеансом предназначены параметры меню Сеанс:

Сеанс → Создать (Ctrl+N)

Открывает новое окно KAlgebra.

Сеанс → Полноэкранный режим (Ctrl+Shift+F)

Включение и отключение полноэкранного режима для окна KAlgebra. Полноэкранный режим также возможно включать и отключать с помощью кнопки в верхнем правом углу окна KAlgebra.

Сеанс → Выход (Ctrl+Q)

Завершает работу программы.

Глава 2

Синтаксис

В KAlgebra для ввода пользовательских функций используется интуитивно понятный алгебраический синтаксис, похожий на тот, который применяется в большинстве современных графических калькуляторов. В этом разделе перечислены основные встроенные операции, доступные в KAlgebra. Автор KAlgebra взял за основу синтаксис программ Maxima и Maple, что упростит работу знакомым с ними пользователям.

Внутренняя служба преобразует введённые пользователем выражения в формат MathML — пользователи, которые обладают начальными знаниями о поддерживаемых MathML возможностях, при желании смогут разобраться во внутреннем устройстве KAlgebra.

Вот список доступных в этой версии операций:

- + - * / : Сложение, вычитание, умножение и деление.
- ^, ** : Возведение в степень. Возможно использовать символы Unicode — ² и так далее. Операция взятия корня обратна операции возведения в степень: a**(1/b)
- -> : лямбда. Способ указать одну или несколько свободных переменных, которые будут связаны в функции. Например, в выражении length:=(x,y)->(x*x+y*y)^0.5 лямбда-оператор позволяет обозначить, что переменные x и y будут связаны при использовании функции length.
- x=a..b : Эта конструкция используется, если нужно задать диапазон значений переменной (связанная переменная + ограничение снизу + ограничение сверху). Это означает, что x изменяется в интервале от a до b.
- () : Скобки используются для явного задания приоритета вычислений.
- abc(параметры) : Функции. В роли abc может быть згнак оператора или имя функции.
- := : Определение. Задаёт значение переменной. Допустимы выражения наподобие x:=3, x:=y (даже если y не определена), perimeter:=r->2*pi*r.
- ? : Условный оператор для задания кусочно-заданных функций в KAlgebra. Другими словами, это способ указания условия «if», «elseif», «else». Если указать условие перед знаком «?», то следующее за ним значение будет использовано только в том случае, если условие выполняется, а если перед «?» не будет условия, будет использовано предыдущее условие. Пример: piecewise { x=0 ? 0, x=1 ? x+1, ? x**2 }
- { } : Блок MathML. Позволяет указывать произвольный код на MathML внутри. Полезно для работы с кусочно-заданными функциями.
- = > >= < <= : Операции сравнения «равно», «больше», «больше или равно», «меньше», «меньше или равно».

Зачем вообще использовать блок MathML? Ответ прост. С его помощью возможно выполнять операции наподобие cos(), sin() и любые тригонометрические функции, sum() и product(). Характер самой функции не важен. Возможно использовать plus() (сложение), times() (умножение) и любые другие функции, которым соответствует определённый оператор. Также реализованы булевские функции, что позволяет выполнять операции наподобие or(1,0,0,0,0).

Глава 3

Использование калькулятора

Калькулятор KAlgebra обладает очень широкими возможностями. Ввод выражений доступен в режиме Вычислить или Упростить (в зависимости от варианта, который выбран в меню Калькулятор).

В режиме упрощения KAlgebra упрощает выражение даже при наличии неопределённой переменной. В режиме вычисления KAlgebra производит вычисление и отображает ошибку при обнаружении неопределённой переменной.

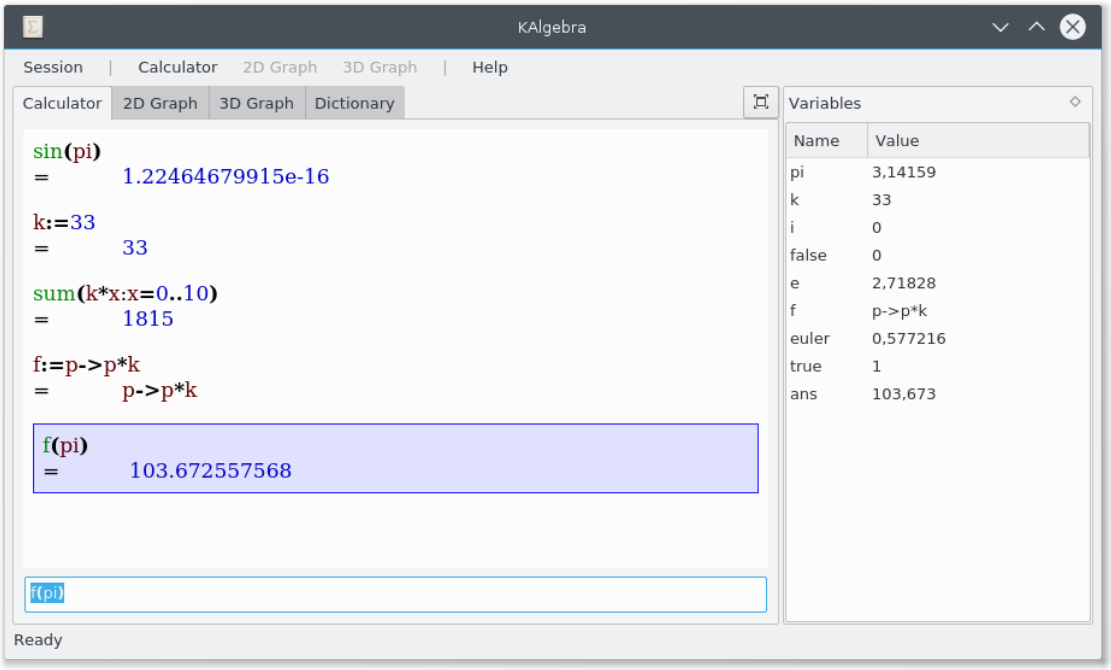
В области калькулятора показаны уравнения, которые были введены пользователем, и результаты вычислений, а в закреплённой области в правой части окна — все объявленные переменные. Двойной щелчок по переменной позволяет открыть диалог редактирования её значения (способ «обойти» журнал).

Переменная «ans» является особой: при вводе каждого нового выражения её значение будет заменяться на последний результат.

Далее приводятся примеры функций, которые возможно указывать в поле ввода окна калькулятора:

- sin(pi)
- k:=33
- sum(k*x : x=0..10)
- f:=p->p*k
- f(pi)

Далее показано окно калькулятора после ввода приведённых ранее примеров выражений:



Параметры меню Калькулятор позволяют управлять выполнением последовательности вычислений:

Калькулятор → Открыть сценарий... (Ctrl+L)

Последовательно выполняет инструкции из файла. Подходит для определения библиотек или возобновления работы.

Калькулятор → Последние сценарии

Отображает вложенное меню, позволяющее выбрать один из недавно использованных сценариев.

Калькулятор → Сохранить сценарий... (Ctrl+G)

Сохраняет инструкции, набранные пользователем с начала работы для возможности их повторного использования. Генерирует текстовые файлы, легко редактируемые любым текстовым редактором (например, Kate).

Калькулятор → Экспорт журнала... (Ctrl+S)

Сохраняет журнал с результатами в файл HTML для распечатки или публикации.

Калькулятор → Вставить ответ... (F3)

Вставляет переменную «ans», упрощая возможность повторного использования предыдущих значений.

Калькулятор → Вычислить

Переключатель для параметра Режим выполнения на «Вычислить».

Калькулятор → Упростить

Переключатель для параметра Режим выполнения на «Упростить»

Глава 4

Двумерные графики

Чтобы добавить новый двумерный график в KAlgebra, перейдите на вкладку Плоский график и нажмите кнопку Добавить для добавления новой функции. После этого фокус будет перенесён в поле ввода, в котором возможно указать функцию.

4.1 Синтаксис

Если следует использовать простую форму задания функций f(x), не требуется отдельно задавать её, но если следует определить f(y) или функцию в полярных координатах, необходимо указать y-> и q-> в качестве связанных переменных.

Примеры:

- sin(x)
- x²
- y->sin(y)
- q->3*sin(7*q)
- t->vector{sin t, t**2}

После ввода функции нажмите кнопку OK для просмотра графика в главном окне.

4.2 Возможности

Возможно начертить несколько графиков в одном окне. Воспользуйтесь кнопкой Добавить в режиме списка. Для каждого из графиков возможно указать цвет.

Размер и расположение области просмотра возможно изменять с помощью мыши. Колесо мыши позволяет увеличивать и уменьшать область просмотра. Если выделить часть области с помощью левой кнопки мыши, она будет растянута до размеров области просмотра. Для перемещения области просмотра используются клавиши со стрелками.

> Замечание
> Область просмотра двумерных графиков возможно явно определить на вкладке Область просмотра вкладки Плоский график.

На вкладке Список в правой нижней части окна доступна вкладка Редактирование: двойной щелчок позволяет изменить или удалить её, а снятие или установка флажка рядом с названием функции — показать или скрыть её.

В меню Плоский график доступны следующие параметры:

- Сетка: Показать или скрыть сетку
- Сохранять пропорции: Сохранять пропорции при увеличении
- Сохранить: Сохранить (Ctrl+S) график в файл изображения
- Увеличить/уменьшить: Увеличить (Ctrl++) или уменьшить (Ctrl+-) масштаб
- Фактический размер: Сбросить масштаб в изначальное значение
- Разрешение: Далее следует список переключателей для выбора разрешения для графиков

Ниже представлено окно программы, в котором курсор пользователя находится на самом правом корне функции, sin(1/x). При построении графика было выбрано очень высокое разрешение (так как частота колебаний функций увеличивается рядом с началом координат). Также показан «живой курсор» — если навести курсор на какое-либо место области просмотра графика, в нижнем левом углу окна отобразятся соответствующие координаты x и y. По месту расположения «живого курсора» строится «касательная» к графику функции.

Глава 5

Трёхмерные графики

Чтобы построить трёхмерный график с помощью KAlgebra, перейдите на вкладку Объёмный график. Внизу будет доступно поле ввода, куда и следует ввести функцию. В этой версии нельзя определить третью координату, Z. Поддерживаются только функции, которые явно зависят только от x и y, например: (x,y)->x*y, где z=x*y.

Примеры:

- (x,y)->sin(x)*sin(y)
- (x,y)->x/y

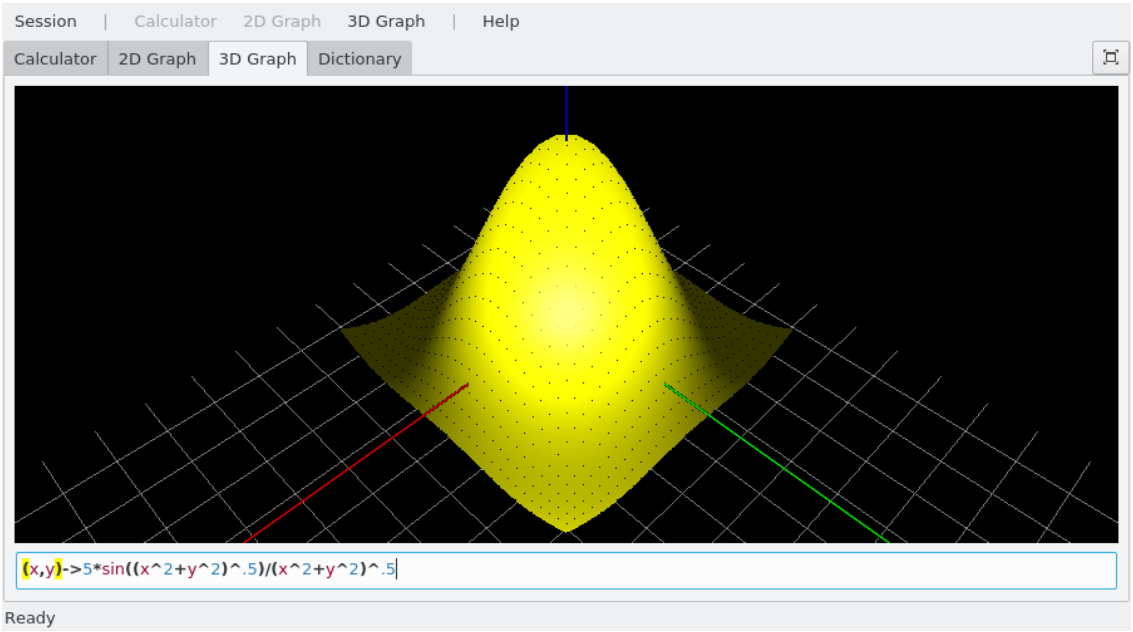
Размер и расположение области просмотра возможно изменять с помощью мыши. Колесо мыши позволяет увеличивать и уменьшать область просмотра. Чтобы вращать график, удерживайте нажатой левую кнопку мыши при её перемещении.

Клавиши со стрелками влево и вправо проворачивают график вокруг оси z, клавиши со стрелками вверх и вниз поворачивают его вокруг горизонтальной оси. Нажмите клавишу W, чтобы увеличить масштаб графика, или клавишу S, чтобы уменьшить его.

Пункты в меню Объёмный график:

- Сохранить: Сохранить (Ctrl+S) график в файл изображения или в поддерживаемый документ.
- Сбросить масштаб: Сбросить масштаб в изначальное значение, установленное в меню Объёмный график
- Для построения графика доступны такие стили как Пунктир, Штрих и Сплошная в меню Объёмный график

Ниже представлена так называемая функция «сомбреро». Её график показан в трёхмерном виде.

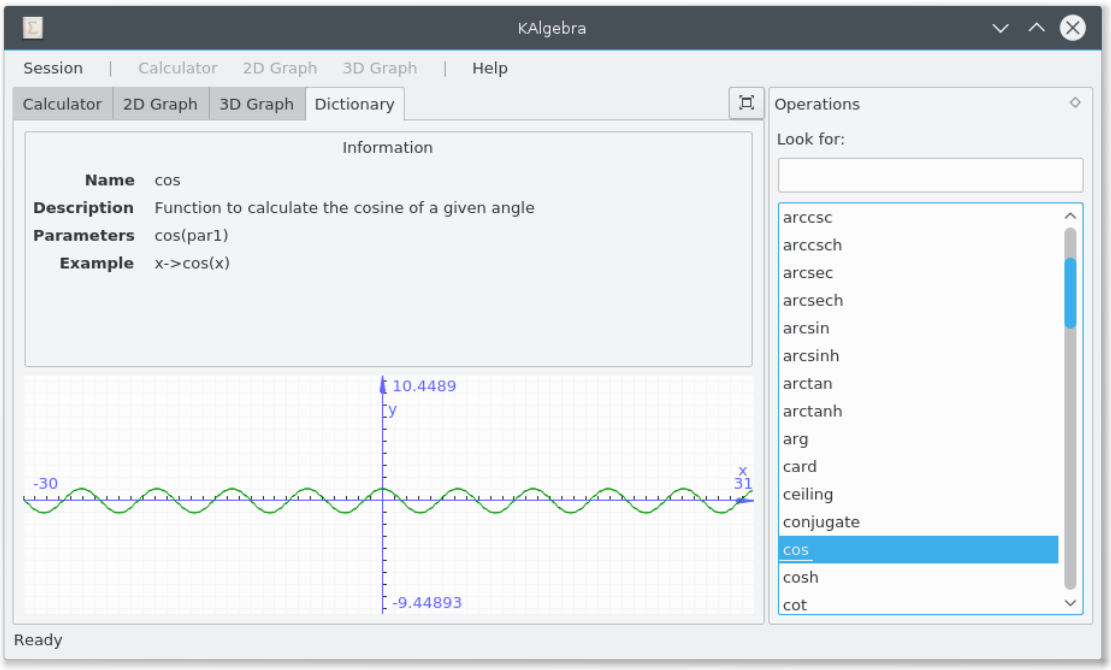

Session | Calculator 2D Graph 3D Graph | Help
Calculator 2D Graph 3D Graph Dictionary
(x,y)->5*sin((x^2+y^2)^.5)/(x^2+y^2)^.5
Ready

Глава 6

Словарь

Словарь является сборником всех доступных действий KAlgebra. Он позволяет найти и просмотреть определение операции и её входные параметры.

Ниже представлено окно KAlgebra с выполнением поиска функции косинуса в словаре.



Глава 7

Команды, поддерживаемые KAlgebra

7.1 plus

- Название: plus
- Описание: сложение
- Использование: plus(... параметры, ...)
- Пример: x->x+2

7.2 times

- Название: times
- Описание: умножение
- Использование: times(... параметры, ...)
- Пример: x->x*2

7.3 minus

- Название: minus
- Описание: вычитание. Вычесть все значения из первого.
- Использование: minus(... параметры, ...)
- Пример: x->x-2

7.4 divide

- Название: divide
- Описание: деление
- Использование: divide(... параметры, ...)
- Пример: x->x/2

7.5 quotient

- Название: quotient
- Описание: частное
- Использование: quotient(... параметры, ...)
- Пример: x->quotient(x, 2)

7.6 power

- Название: power
- Описание: степень
- Использование: power(par1, par2)
- Пример: x->x^2

7.7 root

- Название: root
- Описание: корень
- Использование: root(par1, par2)
- Пример: x->root(x, 2)

7.8 factorial

- Название: factorial
- Описание: факториал. factorial(n)=n!
- Использование: factorial(par1)
- Пример: x->factorial(x)

7.9 and

- Название: and
- Описание: логическое И
- Использование: and(... параметры, ...)
- Пример: x->piecewise { and(x>-2, x<2) ? 1, ? 0 }

7.10 or

- Название: or
- Описание: логическое ИЛИ
- Использование: or(... параметры, ...)
- Пример: x->piecewise { or(x>2, x>-2) ? 1, ? 0 }

7.11 xor

- Название: xor
- Описание: логическое исключающее ИЛИ
- Использование: xor(... параметры, ...)
- Пример: x->piecewise { xor(x>0, x<3) ? 1, ? 0 }

7.12 not

- Название: not
- Описание: логическое отрицание
- Использование: not(par1)
- Пример: x->piecewise { not(x>0) ? 1, ? 0 }

7.13 gcd

- Название: gcd
- Описание: наибольший общий делитель
- Использование: gcd(... параметры, ...)
- Пример: x->gcd(x, 3)

7.14 lcm

- Название: lcm
- Описание: наименьшее общее кратное
- Использование: lcm(... параметры, ...)
- Пример: x->lcm(x, 4)

7.15 rem

- Название: rem
- Описание: остаток от деления
- Использование: rem(par1, par2)
- Пример: x->rem(x, 5)

7.16 factorof

- Название: factorof
- Описание: возвращает истину, если par1 делится на par2 без остатка
- Использование: factorof(par1, par2)
- Пример: x->factorof(x, 3)

7.17 max

- Название: max
- Описание: возвращает максимальный из параметров
- Использование: max(... параметры, ...)
- Пример: x->max(x, 4)

7.18 min

- Название: min
- Описание: возвращает минимальный из параметров
- Использование: min(... параметры, ...)
- Пример: x->min(x, 4)

7.19 lt

- Название: lt
- Описание: если par1 меньше par2, возвращает истину. lt(a,b)=a<b
- Использование: lt(par1, par2)
- Пример: x->piecewise { x<4 ? 1, ? 0 }

7.20 gt

- Название: gt
- Описание: если par1 больше par2, возвращает истину. gt(a,b)=a>b
- Использование: gt(par1, par2)
- Пример: x->piecewise { x>4 ? 1, ? 0 }

7.21 eq

- Название: eq
- Описание: если par1 равно par2, возвращает истину. eq(a,b)=a=b
- Использование: eq(par1, par2)
- Пример: x->piecewise { x=4 ? 1, ? 0 }

7.22 neq

- Название: neq
- Описание: если par1 не равно par2, возвращает истину. neq(a,b)=a≠b
- Использование: neq(par1, par2)
- Пример: x->piecewise { x!=4 ? 1, ? 0 }

7.23 leq

- Название: leq
- Описание: если par1 меньше или равно par2, возвращает истину. leq(a,b)=a⩽b
- Использование: leq(par1, par2)
- Пример: x->piecewise { x<=4 ? 1, ? 0 }

7.24 geq

- Название: geq
- Описание: если par1 больше или равно par2, возвращает истину. geq(a,b)=a⩾b
- Использование: geq(par1, par2)
- Пример: x->piecewise { x>=4 ? 1, ? 0 }

7.25 implies

- Название: implies
- Описание: логическое следование
- Использование: implies(par1, par2)
- Пример: x->piecewise { implies(x<0, x<3) ? 1, ? 0 }

7.26 approx

- Название: approx
- Описание: приблизительное сравнение. approx(a)=a±n
- Использование: approx(par1, par2)
- Пример: x->piecewise { approx(x, 4) ? 1, ? 0 }

7.27 abs

- Название: abs
- Описание: абсолютное значение. abs(n)=|n|
- Использование: abs(par1)
- Пример: x->abs(x)

7.28 floor

- Название: floor
- Описание: округление числа до ближайшего целого в меньшую сторону. floor(n)=⌊n⌋
- Использование: floor(par1)
- Пример: x->floor(x)

7.29 ceiling

- Название: ceiling
- Описание: округление числа до ближайшего целого в большую сторону. ceil(n)=⌈n⌉
- Использование: ceiling(par1)
- Пример: x->ceiling(x)

7.30 sin

- Название: sin
- Описание: функция для вычисления синуса данного угла
- Использование: sin(par1)
- Пример: x->sin(x)

7.31 cos

- Название: cos
- Описание: функция для вычисления косинуса данного угла
- Использование: cosin(par1)
- Пример: x->cos(x)

7.32 tan

- Название: tan
- Описание: функция для вычисления тангенса данного угла
- Использование: tan(par1)
- Пример: x->tan(x)

7.33 sec

- Название: sec
- Описание: функция для вычисления секанса данного угла
- Использование: sec(par1)
- Пример: x->sec(x)

7.34 csc

- Название: csc
- Описание: функция для вычисления косеканса данного угла
- Использование: csc(par1)
- Пример: x->csc(x)

7.35 cot

- Название: cot
- Описание: функция для вычисления котангенса данного угла
- Использование: cot(par1)
- Пример: x->cot(x)

7.36 sinh

- Название: sinh
- Описание: функция для вычисления гиперболического синуса данного угла
- Использование: sinh(par1)
- Пример: x->sinh(x)

7.37 cosh

- Название: cosh
- Описание: функция для вычисления гиперболического синуса данного угла
- Использование: cosh(par1)
- Пример: x->cosh(x)

7.38 tanh

- Название: tanh
- Описание: функция для вычисления гиперболического тангенса данного угла
- Использование: tanh(par1)
- Пример: x->tanh(x)

7.39 sech

- Название: sech
- Описание: функция для вычисления гиперболического секанса данного угла
- Использование: sech(par1)
- Пример: x->sech(x)

7.40 csch

- Название: csch
- Описание: функция для вычисления гиперболического секанса данного угла
- Использование: csch(par1)
- Пример: x->csch(x)

7.41 coth

- Название: coth
- Описание: функция для вычисления гиперболического котангенса данного угла
- Использование: coth(par1)
- Пример: x->coth(x)

7.42 arcsin

- Название: arcsin
- Описание: функция для вычисления арксинуса данного угла
- Использование: arcsin(par1)
- Пример: x->arcsin(x)

7.43 arccos

- Название: arccos
- Описание: функция для вычисления арккосинуса данного угла
- Использование: arccos(par1)
- Пример: x->arccos(x)

7.44 arctan

- Название: arctan
- Описание: функция для вычисления арктангенса данного угла
- Использование: arctan(par1)
- Пример: x->arctan(x)

7.45 arccot

- Название: arccot
- Описание: функция для вычисления арккотангенса данного угла
- Использование: arccot(par1)
- Пример: x->arccot(x)

7.46 arccosh

- Название: arccosh
- Описание: функция для вычисления гиперболического арккосинуса данного угла
- Использование: arccosh(par1)
- Пример: x->arccosh(x)

7.47 arccsc

- Название: arccsc
- Описание: функция для вычисления арккосеканса данного угла
- Использование: arccsc(par1)
- Пример: x->arccsc(x)

7.48 arccsch

- Название: arccsch
- Описание: функция для вычисления гиперболического арккосеканса данного угла
- Использование: arccsch(par1)
- Пример: x->arccsch(x)

7.49 arcsec

- Название: arcsec
- Описание: функция для вычисления арксеканса данного угла
- Использование: arcsec(par1)
- Пример: x->arcsec(x)

7.50 arcsech

- Название: arcsech
- Описание: функция для вычисления гиперболического арксеканса данного угла
- Использование: arcsech(par1)
- Пример: x->arcsech(x)

7.51 arcsinh

- Название: arcsinh
- Описание: функция для вычисления гиперболического арксинуса данного угла
- Использование: arcsinh(par1)
- Пример: x->arcsinh(x)

7.52 arctanh

- Название: arctanh
- Описание: функция для вычисления гиперболического арктангенса данного угла
- Использование: arctanh(par1)
- Пример: x->arctanh(x)

7.53 exp

- Название: exp
- Описание: экспонента (e^x)
- Использование: exp(par1)
- Пример: x->exp(x)

7.54 ln

- Название: ln
- Описание: натуральный логарифм
- Использование: ln(par1)
- Пример: x->ln(x)

7.55 log

- Название: log
- Описание: десятичный логарифм
- Использование: log(par1)
- Пример: x->log(x)

7.56 conjugate

- Название: conjugate
- Описание: сопряжение
- Использование: conjugate(par1)
- Пример: x->conjugate(x*i)

7.57 arg

- Название: arg
- Описание: аргумент
- Использование: arg(par1)
- Пример: x->arg(x*i)

7.58 real

- Название: real
- Описание: функция для вычисления действительной части комплексного числа
- Использование: real(par1)
- Пример: x->real(x*i)

7.59 imaginary

- Название: imaginary
- Описание: функция для вычисления мнимой части комплексного числа
- Использование: imaginary(par1)
- Пример: x->imaginary(x*i)

7.60 sum

- Название: sum
- Описание: сумматор
- Использование: sum(par1 : перем=от..до)
- Пример: x->x*sum(t*t:t=0..3)

7.61 product

- Название: product
- Описание: произведение
- Использование: product(par1 : перем=от..до)
- Пример: x->product(t+t:t=1..3)

7.62 diff

- Название: diff
- Описание: дифференцирование
- Использование: diff(par1 : перем)
- Пример: x->(diff(x^2:x))(x)

7.63 card

- Название: card
- Описание: кардинальное число
- Использование: card(par1)
- Пример: x->card(vector { x, 1, 2 })

7.64 scalarproduct

- Название: scalarproduct
- Описание: скалярное произведение
- Использование: scalarproduct(... параметры, ...)
- Пример: x->scalarproduct(vector { 0, x }, vector { x, 0 })[1]

7.65 selector

- Название: selector
- Описание: функция, равная одному из своих аргументов на всех наборах их значений
- Использование: selector(par1, par2)
- Пример: x->scalarproduct(vector { 0, x }, vector { x, 0 })[1]

7.66 union

- Название: union
- Описание: объединяет несколько объектов одного типа
- Использование: union(... параметры, ...)
- Пример: x->union(list { 1, 2, 3 }, list { 4, 5, 6 })[rem(floor(x), 5)+3]

7.67 forall

- Название: forall
- Описание: для всех
- Использование: forall(par1 : перем)
- Пример: x->piecewise { forall(t:t@list { true, false, false }) ? 1, ? 0 }

7.68 exists

- Название: exists
- Описание: существование
- Использование: exists(par1 : перем)
- Пример: x->piecewise { exists(t:t@list { true, false, false }) ? 1, ? 0 }

7.69 map

- Название: map
- Описание: применение функции к каждому элементу списка
- Использование: map(par1, par2)
- Пример: x->map(x->x+x, list { 1, 2, 3, 4, 5, 6 })[rem(floor(x), 5)+3]

7.70 filter

- Название: filter
- Описание: удаление всех элементов, которые не соответствуют условию
- Использование: filter(par1, par2)
- Пример: x->filter(u->rem(u, 2)=0, list { 2, 4, 3, 4, 8, 6 })[rem(floor(x), 5)+3]

7.71 transpose

- Название: transpose
- Описание: транспонирование
- Использование: transpose(par1)
- Пример: x->transpose(matrix { matrixrow { 1, 2, 3, 4, 5, 6 } })[rem(floor(x), 5)+3][1]

Глава 8

Авторские права и лицензия

- Авторские права на программу принадлежат Aleix Pol, 2005–2009.

Авторские права на документацию к программе принадлежат Aleix Pol aleixpol@gmail.com, 2007

Перевод на русский язык: Николай Шафоростовshaforostoff@kde.ru

Дополнения и исправления перевода на русский язык: Максим Воробьёвvmax0770@gmail.com

Дополнения и исправления перевода на русский язык: Олеся Герасименкоtranslation-team@basealt.ru

Дополнения и исправления перевода на русский язык: Мария Шикуноватranslation-team@basealt.ru

Этот документ распространяется на условиях GNU Free Documentation License.

Программа распространяется на условиях лицензии GNU General Public License.